



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04L 29/06		A1	(11) International Publication Number: WO 98/42108
			(43) International Publication Date: 24 September 1998 (24.09.98)
(21) International Application Number: PCT/US98/05532		(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, GW, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).	
(22) International Filing Date: 19 March 1998 (19.03.98)			
(30) Priority Data: 08/821,235 20 March 1997 (20.03.97) US			
(71) Applicant: ERICSSON INC. [US/US]; 7001 Development Drive, P.O. Box 13969, Research Triangle Park, NC 27709 (US).			
(72) Inventors: GRAHAM, D. Greg; 544 Savannah Avenue, Lynchburg, VA 24502 (US). DOIRON, Timothy, J.; 403 N. Clinton, New Athens, IL 62264 (US). SPIVEY, Anthony; 312-A Gouyer Drive, Madison Heights, VA 24572 (US). SADROZINSKI, Peter; Apartment 11, 2937 Rivermont Avenue, Lynchburg, VA 24503 (US). CROUCHER, Russell; 208 Crestview Drive, Forest, VA 24551 (US).			
(74) Agents: MOORE, Stanley, R. et al.; Jenkins & Gilchrist, P.C., Suite 3200, 1445 Ross Avenue, Dallas, TX 75202 (US).			

Published

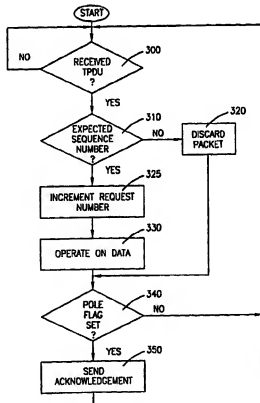
With international search report.

Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: METHOD FOR IMPLEMENTING A TRANSPORT LAYER PROTOCOL FOR WIRELESS PACKET DATA DELIVERY

(57) Abstract

The present invention is a transport layer protocol for data packet delivery in a wireless environment. The method of the present invention incorporates a transport data protocol structure to implement a modified go-back-n automatic repeat request for data packet delivery. The invention further employs a distinct pair of sequence numbers and request numbers for client and server originated data packet transmissions. Several packets can be transmitted by the sender without the need for each packet to be individually acknowledged by the receiver. After n successive packets have been transmitted, the sender polls the receiver for an acknowledgment. The acknowledgment (350) sent by the receiving party contains the request number equal to the sequence number of the packet that the receiver requests to be sent next (325). This number represents the last packet which was received in order plus one (325). Lastly, the present invention provides a method for the client and server to imply that an acknowledgment (350) has been sent to it even though the acknowledgement was lost in transmission.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

-1-

METHOD FOR IMPLEMENTING A TRANSPORT LAYER
PROTOCOL FOR WIRELESS PACKET DATA DELIVERY

BACKGROUND OF THE INVENTION

5 Technical Field of the Invention

The present invention pertains in general to a transport layer protocol and more particularly, to a method for implementing a modified go-back-n automatic repeat request protocol in the transport layer for
10 wireless packet data delivery.

Description of Related Art

The International Standards Organization (ISO) has established a seven layer model regarding communication protocols. The forth layer of this model is referred to
15 as the transport layer and is responsible for packet data delivery between a client and a server. To date, the vast majority of communication using packet data delivery has occurred via a wireline link. A standard solution for providing reliable packet data delivery in the wired
20 environment is the use of Transport Control Protocol (TCP). Transport Control Protocol was designed for wired environments characterized by relatively reliable physical connections between communication devices having large memories and computing power.

25 For various known reasons Transport Control Protocol is unsuited for the harsh environment of wireless packet switched networks. For example, Transport Control Protocol requires a large memory to store both the software for implementing the Transport Control Protocol
30 and the numerous data packets necessary for implementing the Transport Control Protocol's automatic repeat request methodology which allows for the receipt of out-of-order packets. Wireless communication, however, is characterized by the use of radios with small memories and
35 limited computing power as compared to the wired environment. A further problem related to the use of

-2-

Transport Control Protocol in the wireless environment is the large number of acknowledgments required between a client and a server during a communication session. In a wireless environment there is a greater chance for unsuccessful transmissions due to interference and collisions. The numerous acknowledgments required by the Transport Control Protocol leads to increased retransmissions, is inefficient in the wireless environment, and occupies valuable communications resources.

It would be advantageous, therefore, to devise a method for implementing a transport layer protocol in a wireless environment requiring less memory space for receiving data packets and reduced numbers of acknowledgments.

SUMMARY OF THE INVENTION

The present invention employs a multi-field transport data protocol structure for conducting a communication session. A modified go-back-n automatic repeat request is used to provide reliable data packet delivery requiring minimal memory space and computational power. The present invention numbers non-acknowledgment data packets sequentially using sequence numbers. Several packets are then transmitted by the sender without the need for each packet to be individually acknowledged by the receiver. After a given number of packets have been transmitted, the sender polls the receiver for an acknowledgment. A response to the requested acknowledgment includes a request number indicating the sequence number of the packet the receiver wants to be sent next. If all the packets sent are successfully received, the request number will equal the sequence number of the next packet to be sent. Otherwise, the sender begins retransmission starting with the packet having the request number.

-3-

The present invention provides for a separate and distinct pair of sequence numbers and request numbers for client originated data transmissions and server originated data transmissions. Thus, for client originated data transmissions, the client sequentially numbers transmitted data packets with one set of sequence numbers and the server responds with request numbers equal to the sequence number of the data packet next expected to be sent. Conversely, for server originated data transmissions, the server sequentially numbers transmitted data packets with another set of sequence numbers distinct from client originated transmissions and the server responds with request numbers equal to the sequence number of the data packet next expected to be sent.

The present invention further provides a method for the client and server to imply that an acknowledgment has been sent even though the acknowledgment may have been lost in transmission. The client and server imply an acknowledgment if a subsequent packet is received having an expected sequence number.

BRIEF DESCRIPTION OF THE DRAWINGS

A more complete understanding of the method of the present invention may be acquired by reference to the following Detailed Description when taken in conjunction with the accompanying Drawings wherein:

Figure 1 is a Transport Protocol Data Unit for implementing the present invention;

Figure 2 is a sequence diagram for initiating a communication session;

Figure 3 is a flow diagram for implementing the go-back-n automatic repeat request method of the present invention;

Figure 4 is a sequence diagram for transmitting data from a client to a server without the loss of any data;

Figure 5 is a sequence diagram for transmitting data from a client to a server when a packet of data is lost;

-4-

Figure 6 is a sequence diagram for requesting and transmitting data from a server to a client; and

Figure 7 is a sequence diagram for requesting and transmitting data from a server to a client when acknowledgment messages are unsuccessfully transmitted.

DETAILED DESCRIPTION OF THE INVENTION

Referring now to Figure 1, there is illustrated a Transport Protocol Data Unit (TPDU) 100 for implementing the transport protocol of the present invention. The TPDU 100 contains a header 110 and a trailer 120. Included in the TPDU 100 is a data portion 130 which, together with the header 110 and trailer 120, is limited to a maximum of four hundred sixty-four bytes for the entire TPDU 100. The least significant seven bits of the second byte of the TPDU 100 comprises a command/response field 140. The most significant bit of the second byte of the TPDU 100 comprises a response bit (R)150. If the response bit 150 is clear (i.e. 0), then the TPDU 100 is a command TPDU and the command/response field 140 contains either a command or an indication that the TPDU 100 contains data in the data portion 130. The third byte of the TPDU 100 comprises a response field 160.

A command TPDU does not contain a response, and therefore, the response field 160 is set to a zero value. On the other hand, if the response bit 150 is set (i.e. 1), then the TPDU 100 is a response TPDU and the command/response field 140 contains the command to which the TPDU 100 is responding. The response field 160 contains the response to the command indicated in the command/response field 140.

The third bit of the fourth byte of the TPDU 100 comprises a poll bit (P)170. The poll bit 170 is used to request an acknowledgment. When the poll bit 170 is clear (i.e., 0), the receiver continues to receive data without sending an acknowledgment. When the receiver receives a TPDU 100 having the poll bit 170 set (i.e., 1), the

-5-

receiver sends an acknowledgment to the sender. The fifth byte of the TPDU 100 comprises either the most significant eight bits of a session ID 181 or an eight bit value representing a retry time 182. During an initialization command sent by a client to a server the session ID has not yet been defined and the fifth byte of the TPDU 100 contains an eight bit value informing the server of the initial retry timeout value. This value has a five second resolution making the initial retry time in seconds equal to five times the value provided in this field. Upon receiving the initialization command, the server sends an acknowledgment to the client containing the most significant eight bits of the session ID 181 in the fifth byte and the least significant eight bits of the session ID 183 contained in the sixth byte of the TPDU 100. The most significant eight bits of the session ID 181 together with the least significant eight bits of the session ID 183 comprise a sixteen bit value identifying the current communication session between the client and the server. All TPDU's transmitted between the client and the server must contain the session identifier, otherwise, the data will be ignored.

The seventh and eighth byte of the TPDU 100 comprises the sequence number 180 and the ninth and tenth byte of the TPDU 100 comprises the request number 190. Every TPDU contains a unique sequence number identifying that particular TPDU 100. When a communication session between a client and a server is set up, numbers based on the current time of day are selected for client originated sequence number 180 and for the server originated sequence number which is communicated to the server via the request number 190. For the duration of the communication session, each TPDU 100 is identified with sequentially increasing sequence numbers 180.

The last two bytes of the TPDU 100 forming the trailer 120 contain a sixteen bit cyclic redundancy check for the TPDU 100.

-6-

Referring additionally now to Figure 2, there is illustrated a sequence diagram for initiating a communication session. To initiate a communication session between a client 200 and a server 210, the client 200 transmits an initialization TPDU 220. The initialization TPDU 220 includes an initialization command contained within the command/response field 140, a time based sequence number y contained in the sequence number field 180 and an initial retry time out represented by a number in the retry time field 182 which is equal to the number of seconds divided by five. Upon receiving the initialization TPDU 220 the server 210 transmits an acknowledgment TPDU 230 to the client 200 containing an acknowledgment response in the response field 160, the most significant byte 181 and least significant byte 183 of the session ID, the initialization command in the command/response field 140 and a request number y+1 in the request number field 190. The request number y+1 transmitted by the server 210 signifies that the server 210 is expecting to receive the next sequentially numbered TPDU following the initialization TPDU 220 (i.e., the y+1 numbered TPDU). Upon receipt of the acknowledgment TPDU 230 the client 200 proceeds to transmit a data TPDU 240 containing data in the data field 130 and an indication of the presence of data in the command/response field 140. The data TPDU 240 contains a sequence number of y+1 contained in the sequence number field 180.

Referring additionally now to Figure 3, there is illustrated a flow diagram of the present invention for implementing the go-back-n automatic repeat request methodology. After a communication session has been initiated, the party currently receiving data packets listens for a received TPDU (step 300). When a TPDU is received the receiving party compares the sequence number 180 of the received TPDU against the expected sequence number (step 310). The expected sequence number is the next number in sequence following the sequence number of

-7-

a previously, correctly received TPDU. If the sequence number 180 of the TPDU does not match the expected sequence number, the receiving party discards the received packet (step 320). If, on the other hand, the sequence number 180 of the TPDU matches the expected sequence number the receiving party increments the expected sequence number (step 325) and performs an operation on the TPDU (step 330) in accordance with the command contained in the TPDU field 140. After the receiving party has operated on the data (step 330), or discarded a packet containing a sequence number 180 other than the expected sequence number (step 320) the receiving party determines whether the pole bit 170 is set on the received TPDU (step 340). If the pole bit 170 is not set the receiving party returns to continue monitoring for further transmitted packets (step 300). If, on the other hand, the pole bit 170 is set, the receiving party sends an acknowledgment (step 350) to the sending party, and then returns to continue monitoring for transmitted packets (step 300). The acknowledgment sent by the receiving party contains the request number 190 equal to the sequence number 180 of the packet that the receiver requests to be sent next. This number represents the last packet which was received in order plus one. The method described in the flow diagram of Figure 3 applies equally both to client and server originated transmissions.

Referring additionally now to Figure 4, there is illustrated a sequence diagram for transmitting data from a client 200 to a server 210 when no data packets have been lost. After a communication session has been established, the client 200 proceeds to transmit data packets to the server 210. The number of packets p which the client 200 transmits to the server 210 before requesting an acknowledgment is set by the user. The client 200 transmits multiple data packets to the server 210 and eventually the client 200 transmits the $p-1$ data packet 400 having a sequence number 180 of a . Since one

-8-

additional data packet is to be transmitted prior to requesting an acknowledgment, the pole bit 170 of the TPDU 400 is not set. The client 200 then transmits the pth data packet 410 with the pole bit 170 set. Upon receiving the TPDU 410 the server 210 determines that the pole bit 170 is set and transmits an acknowledgment TPDU 420 containing a request number 190 of $a+2$. In this illustration no data packets have been lost and the requested number $a+2$ of the acknowledgment TPDU 420 is equal to the sequence number 180 for the TPDU that should next be transmitted by the client 200.

Referring additionally now to Figure 5, there is illustrated a sequence diagram for transmitting data from a client to a server where a packet of data has been lost. As with Figure 4, the client 200 of Figure 5 has initiated a communication session and is in the process of transmitting data packets to the server 210. The client 200 transmits TPDU 500 containing a sequence number 180 of b . The TPDU 500 is successfully received by the server 210 which increments its expected sequence number by 1 thereby expecting the next TPDU to contain a sequence number 180 of $b+1$. The client 200 transmits data TPDU 510 with a sequence number 180 of $b+1$ which, for whatever reason, is not received by the server 210. The client 200 having no knowledge of the unsuccessful transmission continues with the transmission of TPDU 520 containing a sequence number 180 of $b+2$ which is received by the server 210. Since the sequence number $b+2$ of TPDU 520 does not match the expected sequence number $b+1$, the server 210 discards the packet. The server 210, however, recognizes that the pole bit 170 of TPDU 520 is set and in response generates an acknowledgment TPDU 530 containing a request number 190 of $b+1$ (instead of $b+3$). Upon receiving the acknowledgment TPDU 530, the client 200 begins retransmission of data packets starting with TPDU 510 having a sequence number 180 of $b+1$.

-9-

Referring additionally now to Figure 6, there is illustrated a sequence diagram for transmitting data from a server 210 to a client 200. This diagram includes server 210 originated data transmissions and illustrates the distinction between sequence numbers 180 and request numbers 190 associated with client 200 originated and server 210 originated transmissions.

The client 200 initiates a communication session by transmitting an initialization TPDU 600. The initialization TPDU 600 contains a randomly generated sequence number 180 of y for client 200 originated transmissions and a time based request number 190 of z corresponding to server 210 originated transmissions. The server 210 upon receiving the initialization TPDU 600 transmits an acknowledgment TPDU 610 containing a request number 190 of $y+1$. Upon receiving the acknowledgment TPDU 610 the client 200 transmits a data request TPDU 620 in which the pole bit 170 is set thereby requesting an acknowledgment. Since this TPDU 620 is a client 200 originated TPDU the sequence number 180 is equal to the sequence number y of the previously transmitted TPDU 600 plus 1 or $y+1$. The data request TPDU 620 also includes a request number 190 of z which is the sequence number 180 which the server 210 will use to begin data transmission. Upon receiving the data request 620 the server 610 transmits an acknowledgment TPDU 630 having a request number 190 of $y+2$. The server 210 then begins to transmit the requested data with server 210 originated TPDU transmissions. These transmissions have sequence numbers 180 beginning with z as requested by the client. A data TPDU 640 having a sequence number 180 of z and a request number 190 of $y+2$ is thus transmitted to the client 200 with the pole bit 170 set to request acknowledgment. The data TPDU 640 also contains the client 200 requested data. Upon receiving the data TPDU 640 the client 200 transmits an acknowledgment TPDU 650 having a request number 190 of $z+1$ equal to the next server 210 originated packet

-10-

sequence number 180 expected by the client 200. The next client 200 originated transmission data TPDU 660 has a client originated sequence number 180 of $y+2$ and a request number 190 of $z+1$. As can be seen by the illustration in Figure 6, the appropriate sequence number 180 and request number 190 depends upon whether the transmission and subsequent acknowledgment is a result of client 200 or server 210 originated data packet transmissions.

A problem associated with server 210 originated data transmissions is that the server 210 has no way of recognizing whether an acknowledgment TPDU which it transmits to the client 200 was unsuccessfully received by the client 200. Likewise, a client 200 has no way of recognizing whether an acknowledgment sent by it to the server 210 in response to the server originated data transmissions has been unsuccessfully received by the server 210. To solve this problem the method of the present invention stipulates that an acknowledgment is implied if a packet is subsequently received which has the expected request number 190.

Referring additionally now to Figure 7, there is illustrated a sequence diagram for requesting and transmitting data from a server 210 to a client 200 when acknowledgment messages are unsuccessfully transmitted. In response to the transmission of a data request TPDU 700, similar to data request TPDU 620 of Figure 6, containing a sequence number 180 of $y+1$, a request number 190 of z , and the pole bit 170 set, the server 210 transmits an acknowledgment TPDU 710 having a request number 190 of $y+2$. The acknowledgment TPDU 710, for whatever reason, is not received by the client 200. Nevertheless, the server 210 transmits a data TPDU 720 containing a sequence number 180 of z , a request number 190 of $y+2$, and the pole bit 170 being set. Although the client 200 has not received the acknowledgment TPDU 710 from the server 210, the client 200, following the methodology of the present invention, recognizes that the

-11-

request number $y+2$ of the data TPDU 720 contains the appropriate request number 190 and implies that an acknowledgment TPDU 710 has been sent by the server 210. The client 200, therefore, retains the data TPDU 720 and transmits an acknowledgment 730 having a request number 190 equal to $z+1$ to the server 210. In a similar fashion, this acknowledgment TPDU 730 may be unsuccessfully received by the server 210. Nevertheless, the client 200 transmits another TPDU, which for this example is data TPDU 740 having a sequence number 180 equal to $y+2$ and a request number 190 equal to $z+1$. The server 210, following the methodology of the present invention, recognizes that the request number $z+1$ is the appropriate request number 190 and implies that the client 200 has transmitted the acknowledgment TPDU 730 even though it was not received by the server 210. The stipulation of implying an acknowledgment if a packet is subsequently received which has the expected request number 190 prevents the client 200 and the server 210 from becoming out of synchronization with one another and constantly re-sending data packets to one another.

Unlike Transport Control Protocol, the transport layer protocol of the present invention does not support a selective automatic repeat request methodology and thus, does not support the receipt of out of order packets. In the present invention, packets received out of order are discarded. If, however, a packet is received out of order but contains a request for acknowledgment, an acknowledgment is sent containing the request number equal to the sequence number of the last packet received in order plus one.

Although a preferred embodiment of the method of the present invention has been illustrated in the accompanying Drawings and described in the foregoing Detailed Description, it is understood that the invention is not limited to the embodiment disclosed, but is capable of numerous rearrangements, modifications and substitutions

-12-

without departing from the spirit of the invention as set forth and defined by the following claims.

-13-

WHAT IS CLAIMED IS:

1. A Transport Protocol Data Unit for a wireless packet data delivery comprising:

- 5 a response bit for signifying the Transport Protocol Data Unit as a command or response Transport Protocol Data Unit;
- a poll bit for requesting an acknowledgment;
- a sequence number field for identifying the Transport Protocol Data Unit;
- 10 a request number field for communicating a sequence number of the Transport Protocol Data Unit requested to be next transmitted.

2. A method for data packet delivery between a sender and a receiver, comprising the steps of:

- 15 transmitting a user selected number of data packets, wherein each packet is sequentially numbered and each packet except for a last packet has a poll bit cleared, the last packet having the poll bit set;
- receiving the data packets transmitted by the
- 20 sender;
- detecting the sequence number and the poll bit of each received packet;
- comparing the sequence number of each received packet against an expected sequence number;
- 25 storing data contained within packets when the sequence number of the packet matches the expected sequence number; otherwise
- discarding the data packets.

3. A method for data packet delivery between a sender and a receiver, comprising the steps of:

- 30 transmitting a user selected number of data packets, wherein each packet is sequentially numbered and each packet except for a last packet has a poll bit cleared, the last packet having the poll bit set;
- 35 receiving the data packets transmitted by the sender;

-14-

detecting the sequence number and the poll bit of each received packet;

comparing the sequence number of each received packet against an expected sequence number;

5 storing data contained within packets when the sequence number of the packet matches the expected sequence number;

discarding data packets when the sequence number of the packet does not match the expected sequence number;
10 and

transmitting an acknowledgment in response to the detection of a set poll bit, the acknowledgment including a request number equal to the expected sequence number.

15 4. The method of Claim 3, further including the steps of:

transmitting the data packet having the sequence number equal to the request number transmitted by the receiver in the acknowledgment; and

20 transmitting all subsequent data packets.

5. A method for data packet delivery between a client and a server when a client requested acknowledgment is not received by the client prior to the receipt of a subsequent data packet, comprising the steps of:

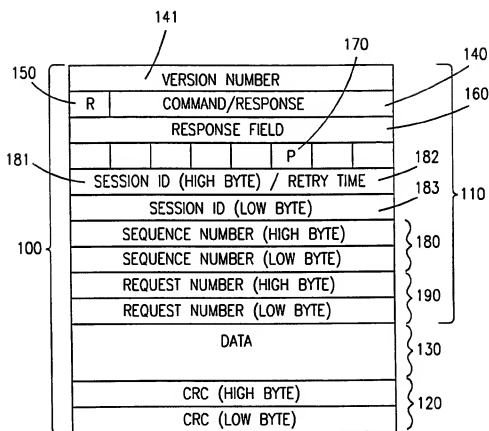
25 receiving a data packet not containing an acknowledgment when an acknowledgment is expected;

detecting a sequence number of the received data packet;

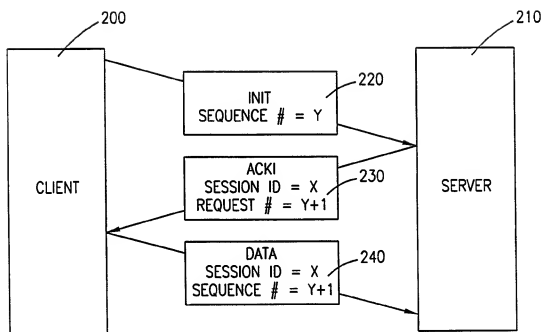
30 comparing the sequence number of the received packet against an expected sequence number; and

implying that an acknowledgment was sent even though not received if the sequence number of the received packet matches the expected sequence number.

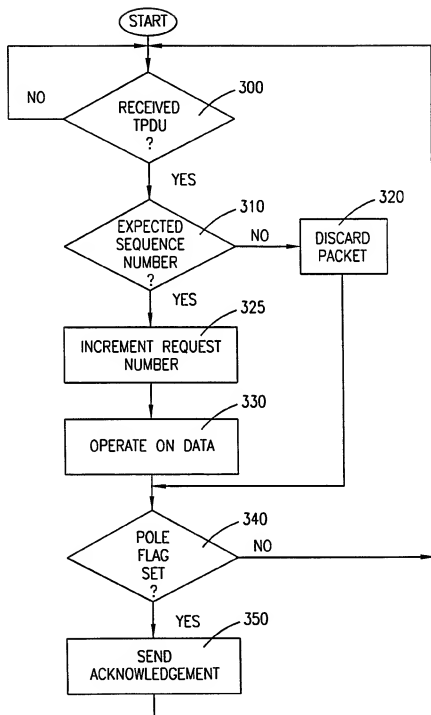
1/7

**FIG. 1**

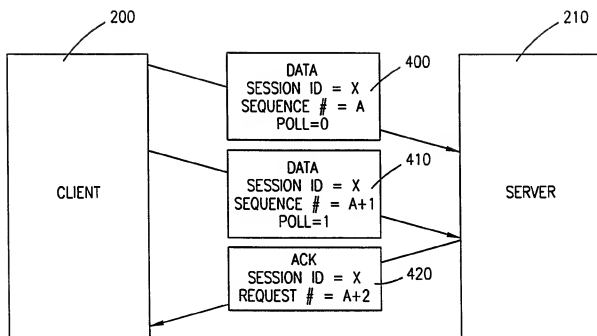
2/7

**FIG. 2**

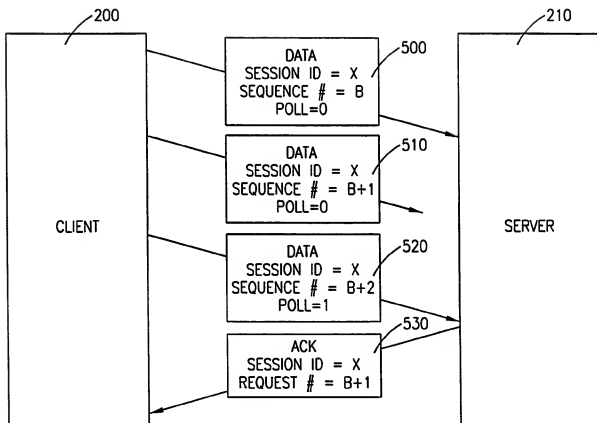
3/7

**FIG. 3**

4/7

**FIG. 4**

5/7

**FIG. 5**

6/7

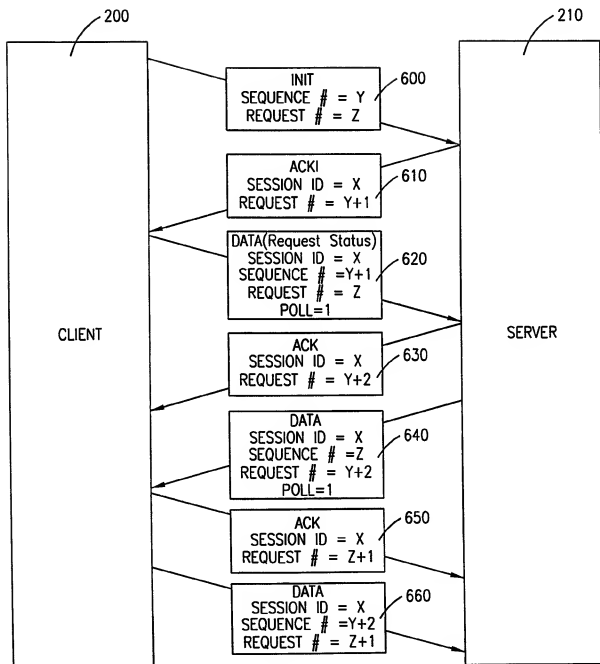
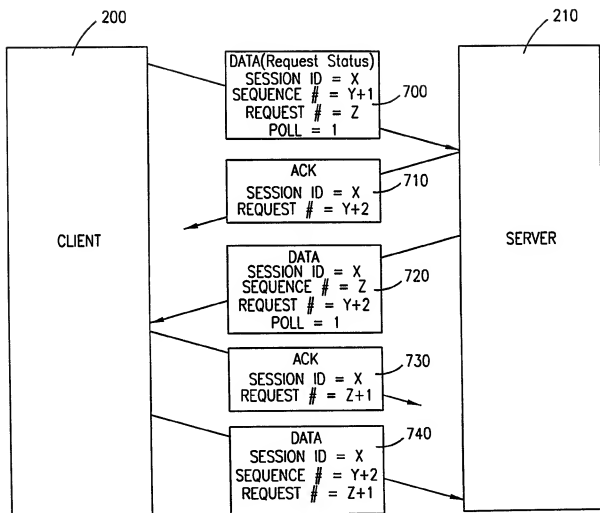


FIG. 6

7/7

**FIG. 7**

INTERNATIONAL SEARCH REPORT

International Application No

PCT/US 98/05532

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 H04L29/06

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 6 H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 448 561 A (KAISER KARL-HEINZ ET AL) 5 September 1995	2
Y	see column 4, line 16-51	1,3
A	see column 5, line 57 - column 6, line 29 see column 7, line 46 - column 8, line 15 see column 19, line 26 - column 22, line 6; figure 11 see column 29, line 50 - column 33, line 22; table 8	4,5
Y	PATENT ABSTRACTS OF JAPAN vol. 015, no. 419 (E-1126), 24 October 1991 & JP 03 175759 A (TOKAI RIYOKAKU TETSUDO KK; OTHERS: 01), 30 July 1991	1,3
A	see abstract; figure --- -/--	2

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"Z" document member of the same patent family

Date of the actual completion of the international search

26 August 1998

Date of mailing of the international search report

03/09/1998

Name and mailing address of the ISA

European Patent Office, P.B. 5618 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Dupuis, H

INTERNATIONAL SEARCH REPORT

Internat. Patent Application No.

PCT/US 98/05532

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 748 096 A (XEROX CORP) 11 December 1996 see page 3, line 16 - page 4, line 8 -----	1-3

INTERNATIONAL SEARCH REPORT

Information on patent family members

Internat. Application No

PCT/US 98/05532

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5448561 A	05-09-1995	DE 4131133 A JP 5235965 A	01-04-1993 10-09-1993
EP 0748096 A	11-12-1996	JP 9016486 A	17-01-1997